2010

Analiza ruchu w sieci

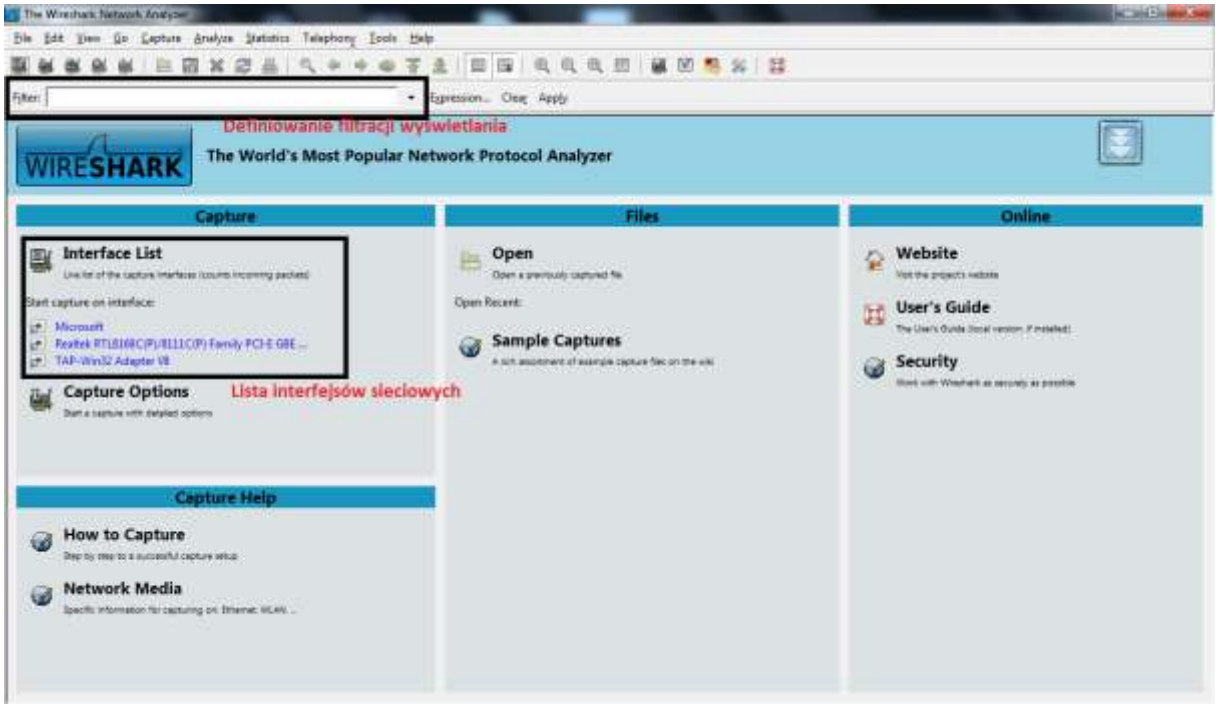
przy pomocy programu Wireshark

[Wpisz tutaj streszczenie dokumentu. Streszczenie jest zazwyczaj krótkim podsumowaniem treści dokumentu. Wpisz tutaj streszczenie dokumentu. Streszczenie jest zazwyczaj krótkim podsumowaniem treści dokumentu.]

Konrad Fierek
Podstawy audytu sieci komputerowych
2010-06-01

1. Pierwsze uruchomienie - rozpoczęcie

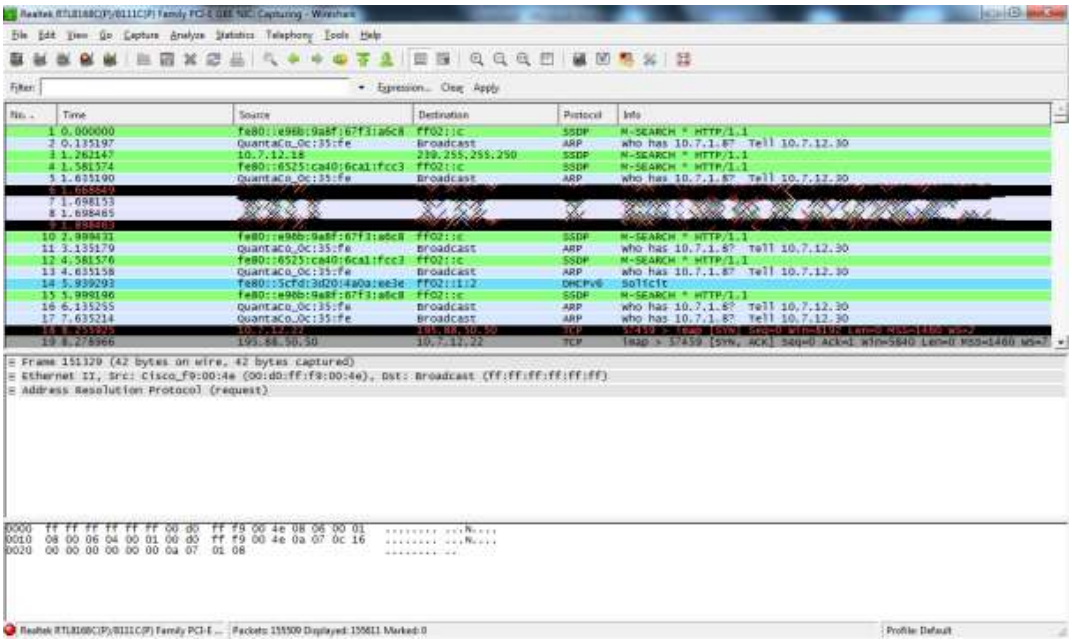
Skanowanie całego ruchu sieciowego



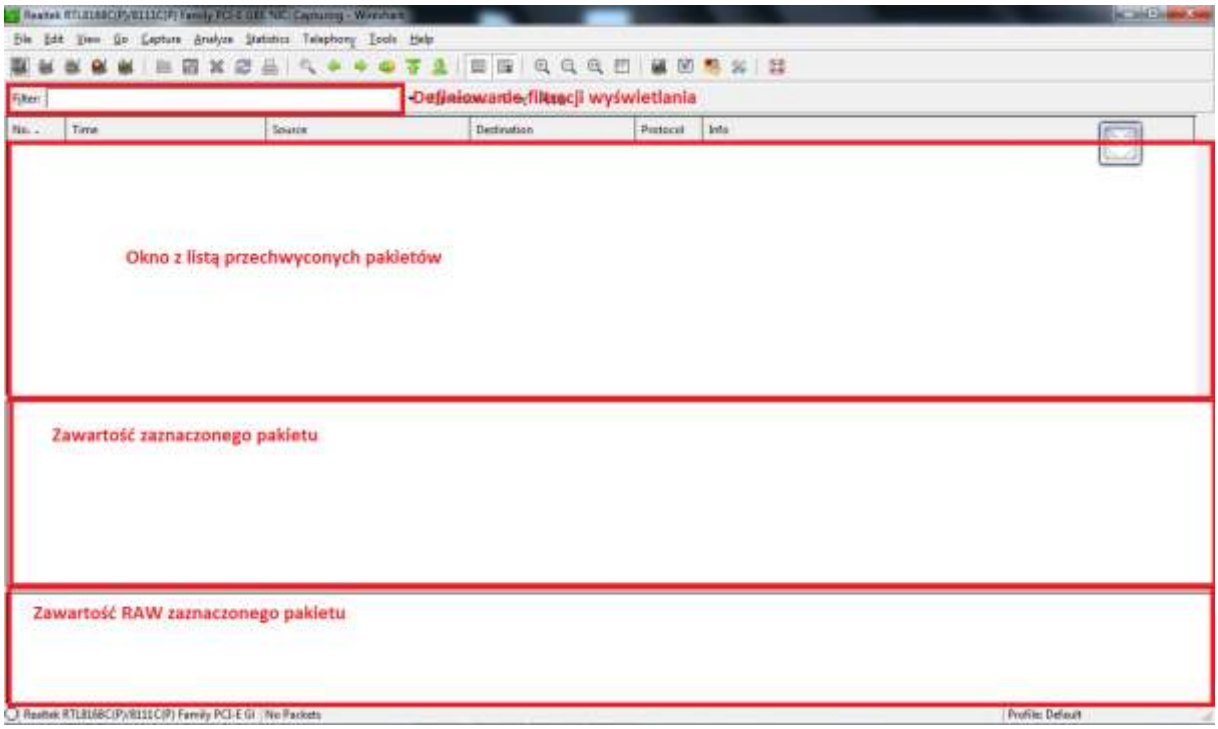
Po uruchomieniu programu wita nas okienko powitalne.

Interesujące nas opcje znajdują się w grupie CAPTURE.

Aby rozpocząć przechwytywanie pakietów, wybieramy z „Interface List” interfejs sieciowy, przy pomocy którego chcemy przechwytywać pakiety.

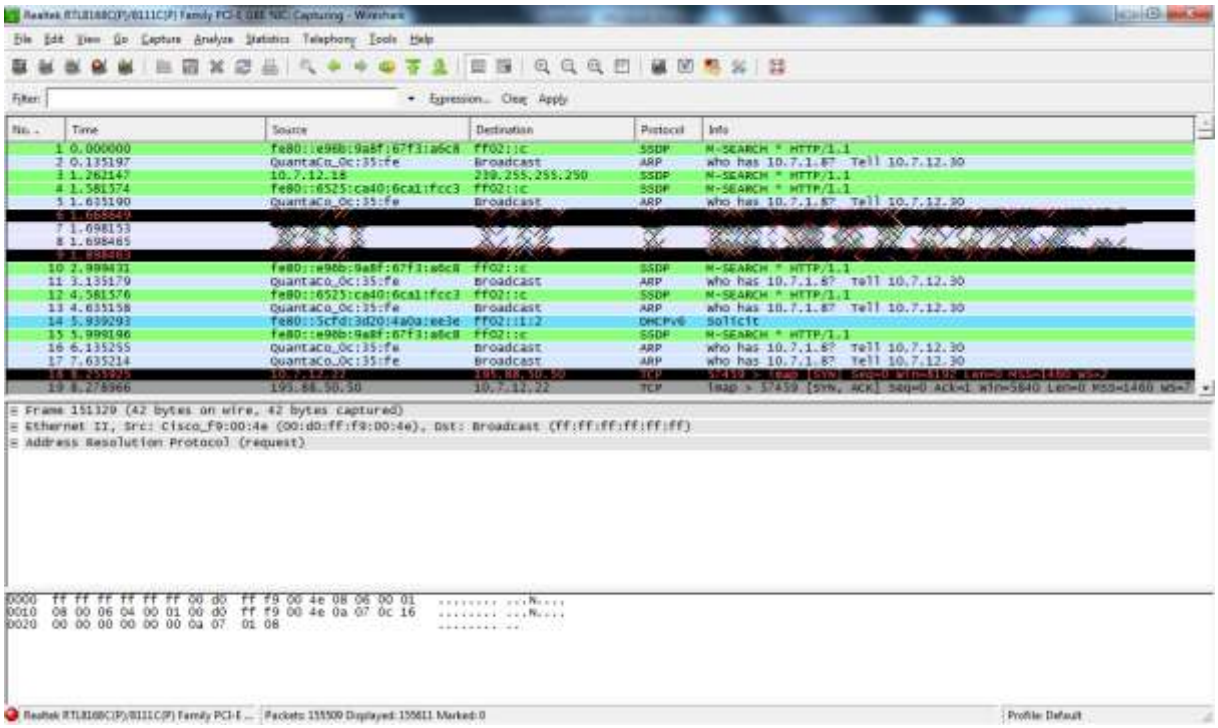


2. Opis podstawowego okna programu



3. Skanowanie ruchu pochodzącego z konkretnego komputera

Po włączeniu przechwytywania pakietów widzimy, jak wielka ilość informacji jest przez naszą sieć transmitowana.



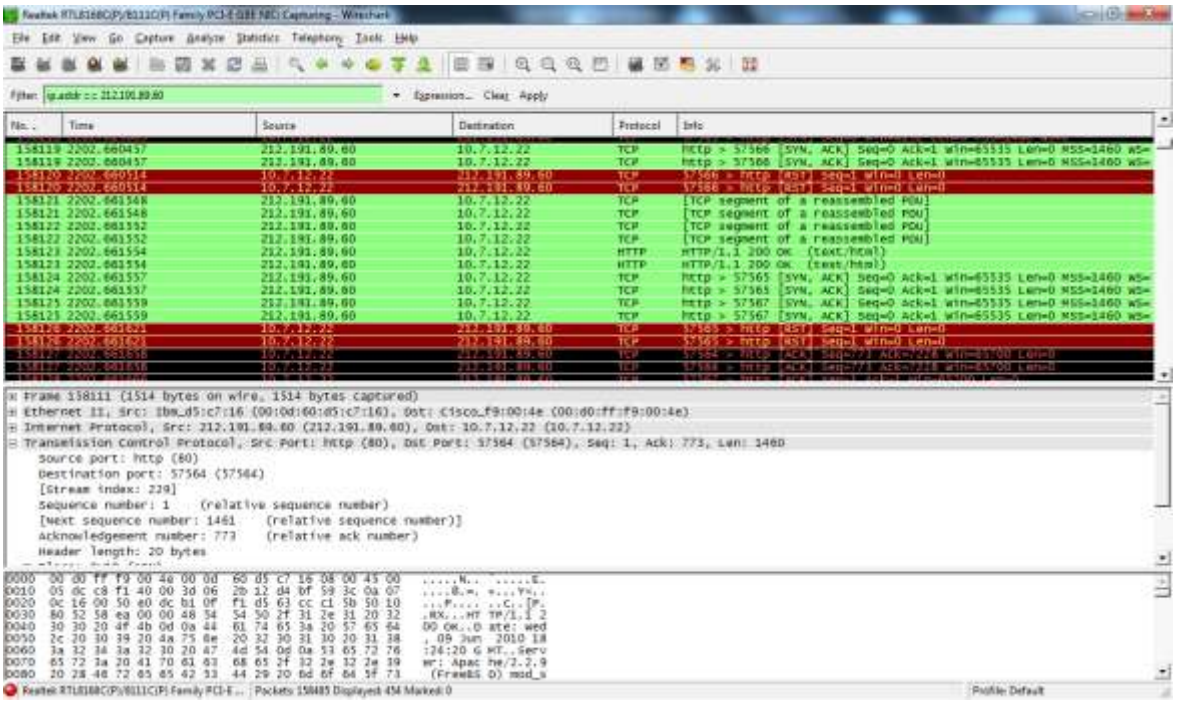
Całą moc pakiet Wireshark ma w tzw. filtrach, które pozwalają zawęzić znacząco wyniki poszukiwań wg ustalonych kryteriów.

Na początku postaramy zawęzić kryteria poszukiwania, aby znajdować pakiety skierowane tylko od/do określonego komputera.

Jeśli zależy nam na tym, alby filtrować komputer po adresie IP, w pole filtru wpisujemy:

```
ip.addr == 1.2.3.4
```

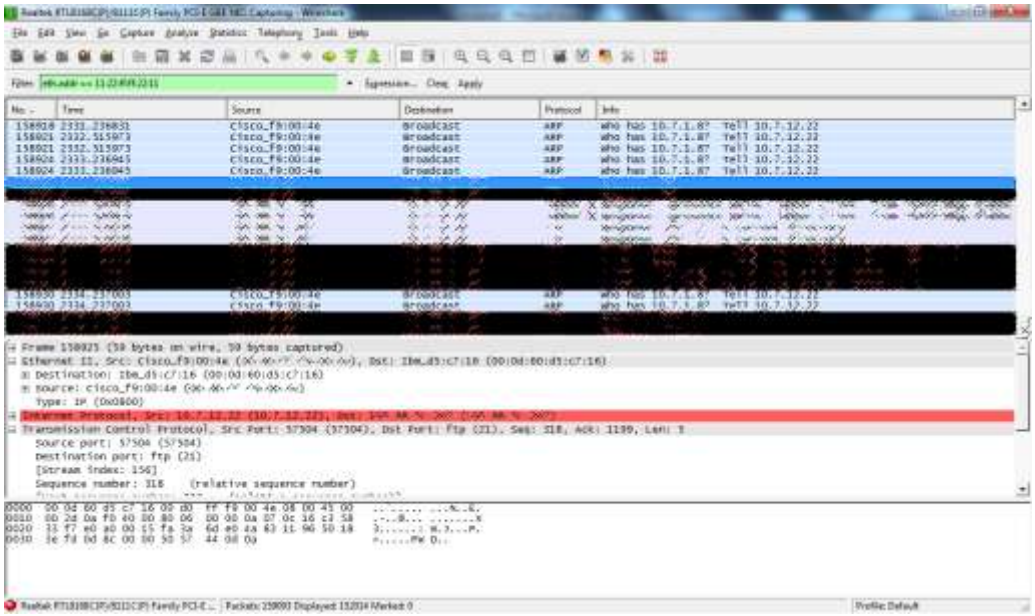
gdzie 1.2.3.4 to adres IP, wg którego chcemy filtrować.



Jeśli zależy nam na odszukaniu hosta po adresie fizycznym MAC, to wpisujemy

```
eth.addr == 11:22:ff:ff:22:11
```

gdzie 11:22:ff:ff:22:11 to adres fizyczny karty sieciowej.



Jeśli szukamy kilku komputerów, możemy posłużyć się spójnikiem *or*

```
eth.addr == 11:22:ff:ff:22:11 or eth.addr == 11:33:ff:ff:33:11
```

Aby zastosować wykluczenie – czyli wszystkie adresy poza określonym, dodajemy z przodu wykrzyknik:

```
    !(ip.addr == 1.2.3.4)
    !(eth.addr == 11:22:ff:ff:22:11)
```

4. Skanowanie konkretnego ruchu

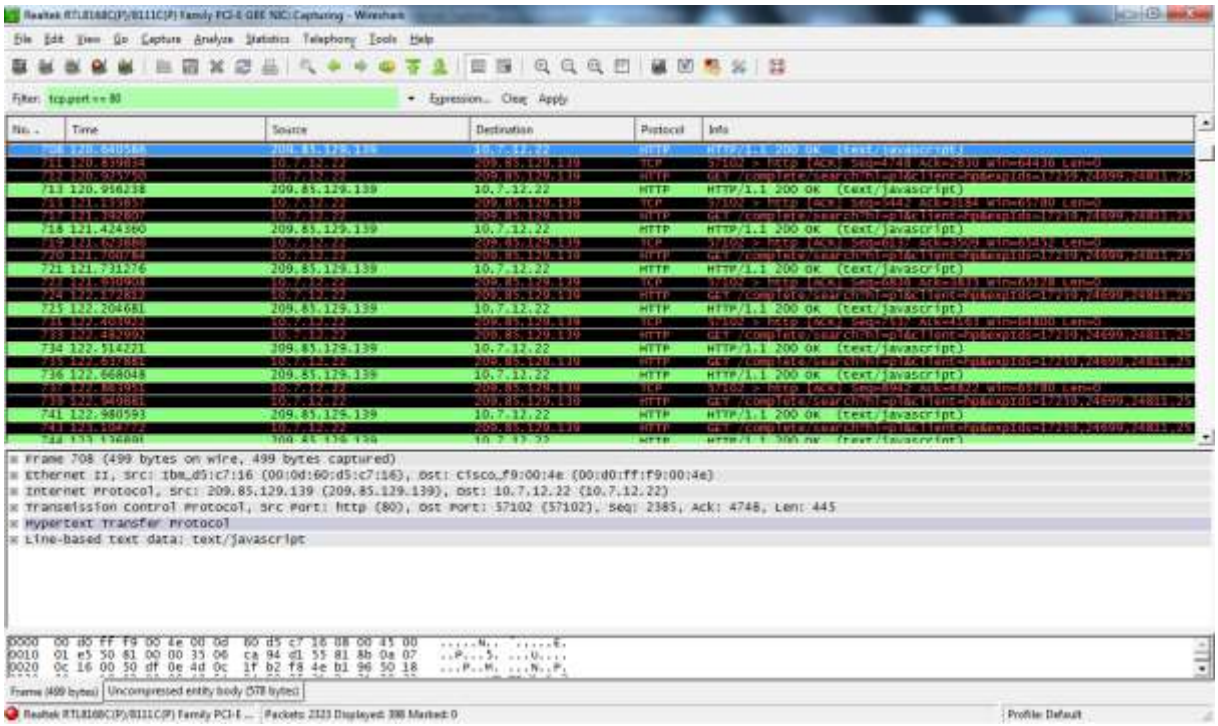
Umiemy już podglądać ruch pochodzący od konkretnego hosta w naszej sieci. Jednak, jak widzimy, ilość rejestrowanych pakietów jest bardzo duża, a co za tym idzie, ich ręczna analiza bardzo czasochłonna i męcząca.

W tym rozdziale zaprezentowane zostaną dodatkowe filtry programu, umożliwiające wyselekcjonowanie nie tylko ruchu na poziomie konkretnego hosta, ale także konkretnego portu, a także nawet pakietów danych konkretnych usług sieciowych.

1) Sesje WWW

Na początek zajmiemy się podejrzeniem tego, jakie strony internetowe są pobierane przez nasz komputer. Gdy spojrzymy do teoretycznej dokumentacji protokołu HTTP wykorzystywanego do transmisji stron internetowych, zauważymy szybko, że działa on na porcie TCP/80. Wyfiltrujmy więc wszelakie pakiety napływające na ten port.

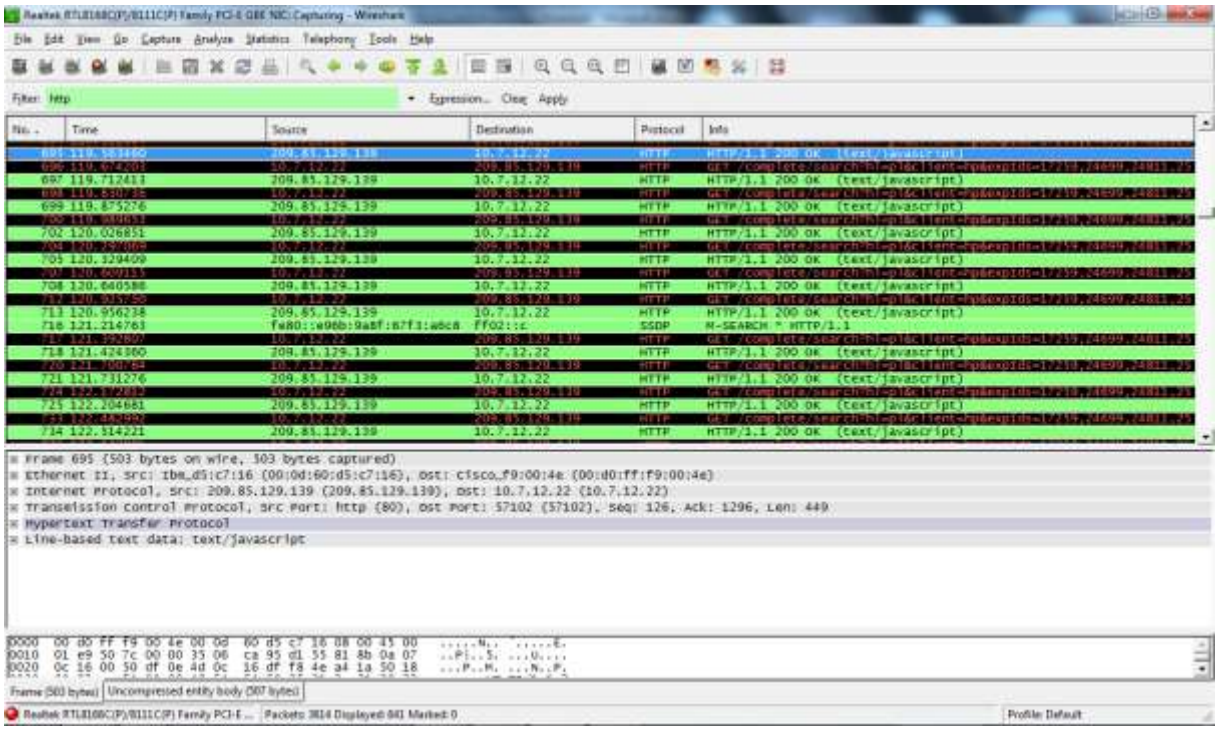
```
tcp.port == 80
```



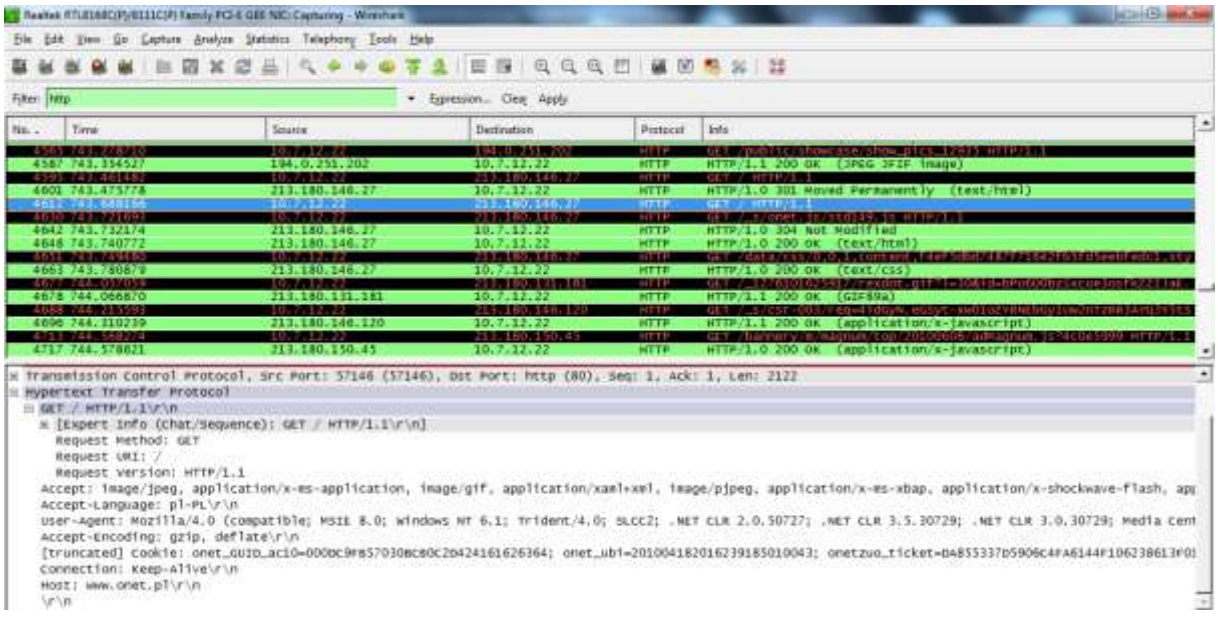
Ukazuje się naszym oczom cały ruch sieciowy na porcie TCP/80. W chwili obecnej łatwiej już nam jest podejrzeć kod źródłowy pobieranych stron. Jednak wśród widocznych pakietów, znajduje się jeszcze sporo pakietów kontrolnych samego połączenia, z naszego punktu widzenia w chwili obecnej zupełnie nieprzydatnych, a tylko zaciemniających użyteczne dla nas dane.

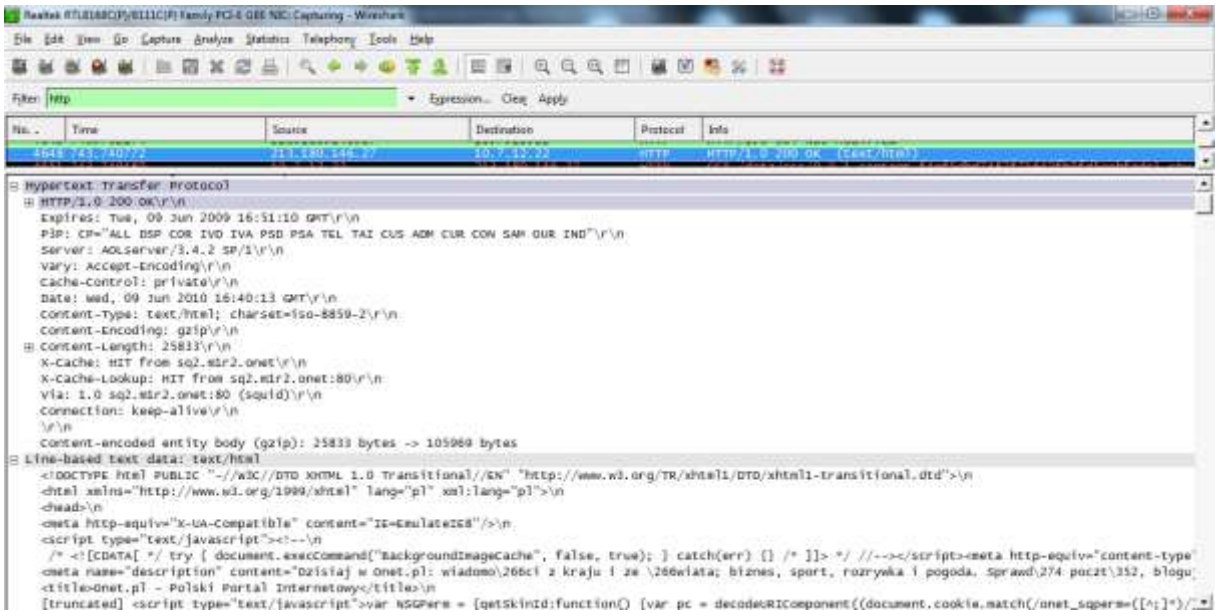
Wykorzystajmy więc dedykowany filtr do analizy protokołu http

```
http
```



Teraz możemy przystąpić do głębszej analizy. Na czarnym tle widoczne są pakiety, zawierające żądania naszego klienta. Na zielonym zaś – odpowiedzi od docelowego serwera. Bezproblemowo obserwujemy wysłane przez nas nagłówki HTTP (rys. 3), jak i też otrzymane od serwera docelowego nagłówki i właściwą treść witryny (rys. 4).





W chwili obecnej jesteśmy w stanie przechwycić wszystkie użyteczne pakiety WWW. Jednak czasami i to kryterium zwraca zbyt dużo wyników. Zawęźmy więc poszukiwania pakietów na te, które skierowane są do serwisu internetowego, który mamy pod lupą.

```
http.host == www.onet.pl
```

Ujrzymy na ekranie wszystkie żądania pobrań strony, skierowane do serwera www.onet.pl

2) Autoryzacja WWW

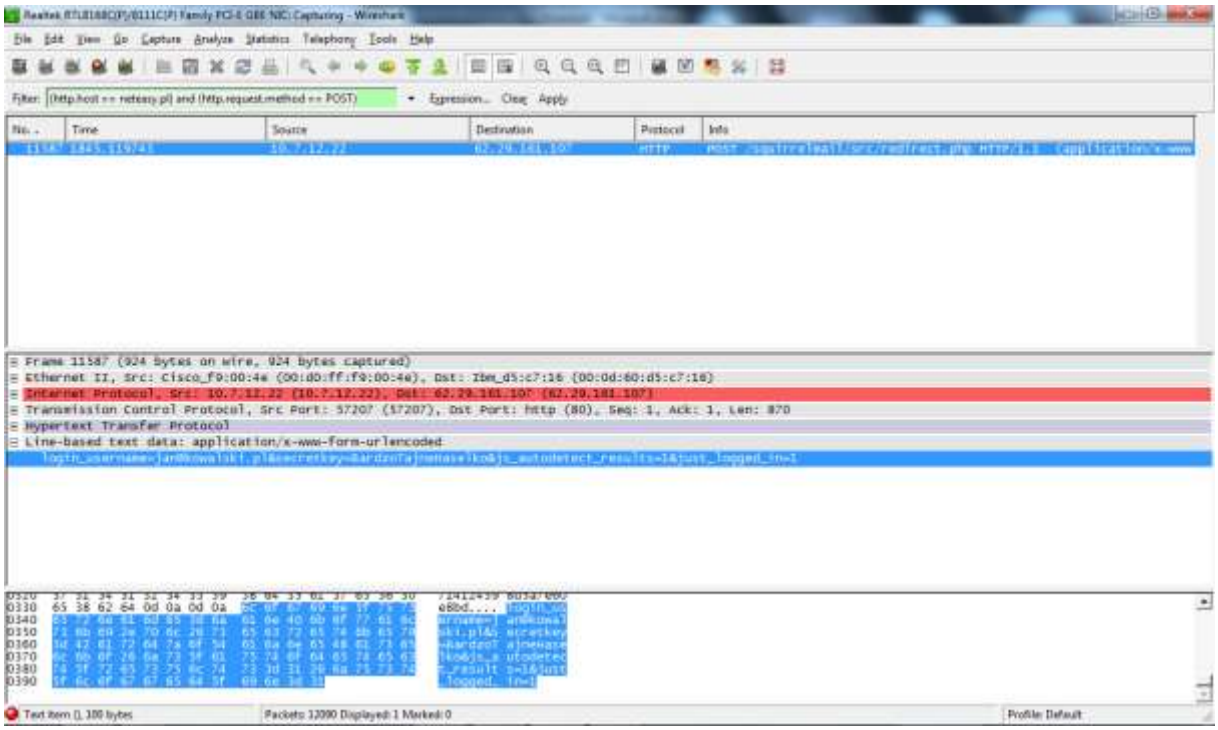
Teraz przystąpimy do trudniejszych spraw ☺ - czyli szybkiej analizy ruchu HTTP pod kątem wprowadzanych danych autoryzacyjnych. Wyróżniamy zasadnicze dwa sposoby poświadczania swoich uprawnień – standardowy formularz HTML, wysyłający metodą POST dane autoryzacyjne, oraz autoryzacja przeglądarką - HTTP/Basic.

a) formularz logowania POST

Przykładowy formularz autoryzacyjny widzimy na rysunku poniżej.

Aby podsłuchać dane, które nasz system wyśle po kliknięciu przycisku Login, ustawiamy następujący filtr:

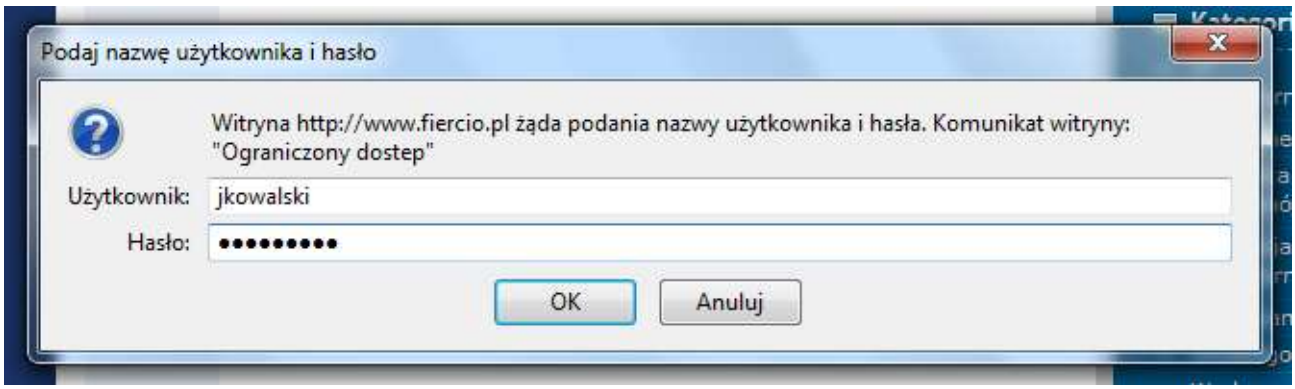
```
(http.host == domena.pl) and (http.request.method == POST)
```



Filtr zwróci nam wtedy wszystkie pakiety, które zawierają w sobie dane wysyłane do serwera metodą POST. Widzimy więc dokładnie podane przez użytkownika w formularzu dane autoryzacyjne, takie jak nazwa użytkownika i tajne hasło.

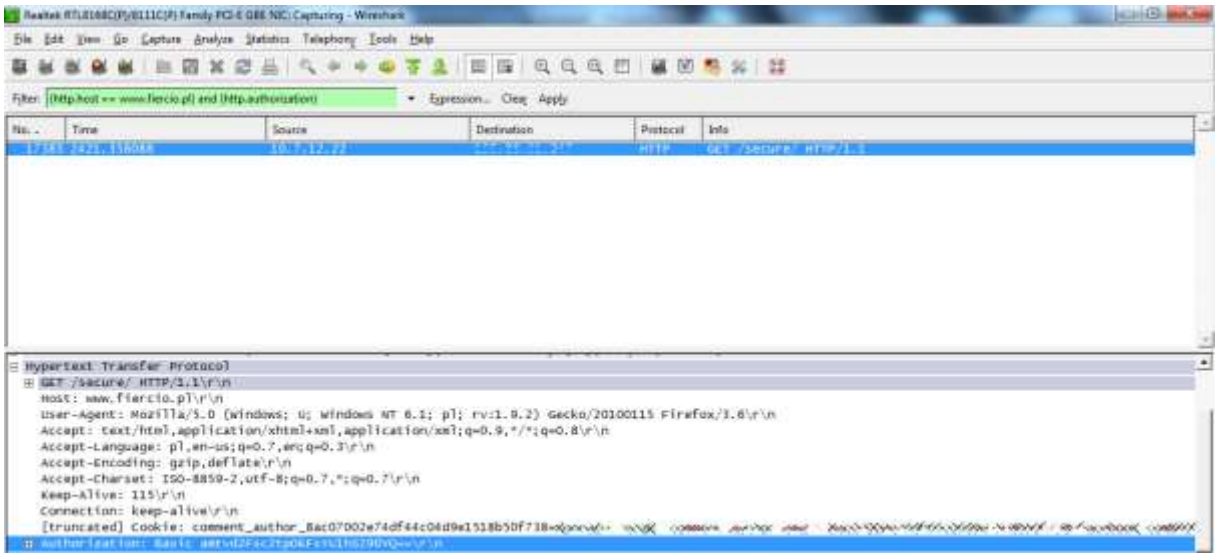
b) autoryzacja HTTP

Drugą z używanych powszechnie metod, jest tzw. autoryzacja HTTP. Po wejściu na chronioną stronę wita nas takie oto okienko przeglądarki:

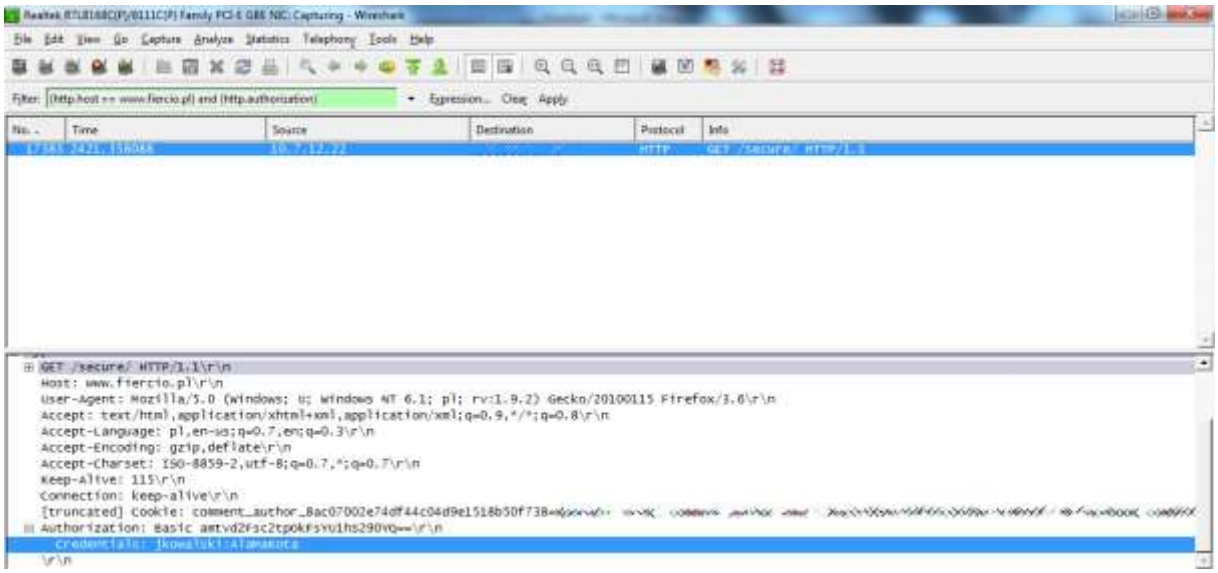


Aby wyfiltrować bezpośrednio dane, ustawiamy filtr

```
(http.host == www.domena.pl) and (http.authorization)
```



Widzimy nasz szukany pakiet. Zawiera on nagłówek Authorization i pewien ciąg znaków, będący frazą autoryzacyjną. Filtr Wiresharka pozwoli nam zdekodować tę frazę do czytelniejszej postaci. Aby zrobić to, klikamy po prostu na plusik przy Autorization. Pokażą nam się wtedy dane autoryzacyjne w postaci NazwaUżytkownika:Hasło.



To tylko częściowy opis filtra HTTP.

Więcej informacji pod adresem: http://www.wireshark.org/docs/dfref/h/http.html

3) Sesje Gadu-Gadu

Gadu-Gadu to jeden z najpopularniejszych komunikatorów w Polsce. Dzięki Panu Arturowi Kołodziejowi możemy ściągnąć wtyczkę do Wiresharka, umożliwiającą analizę protokołu GG.

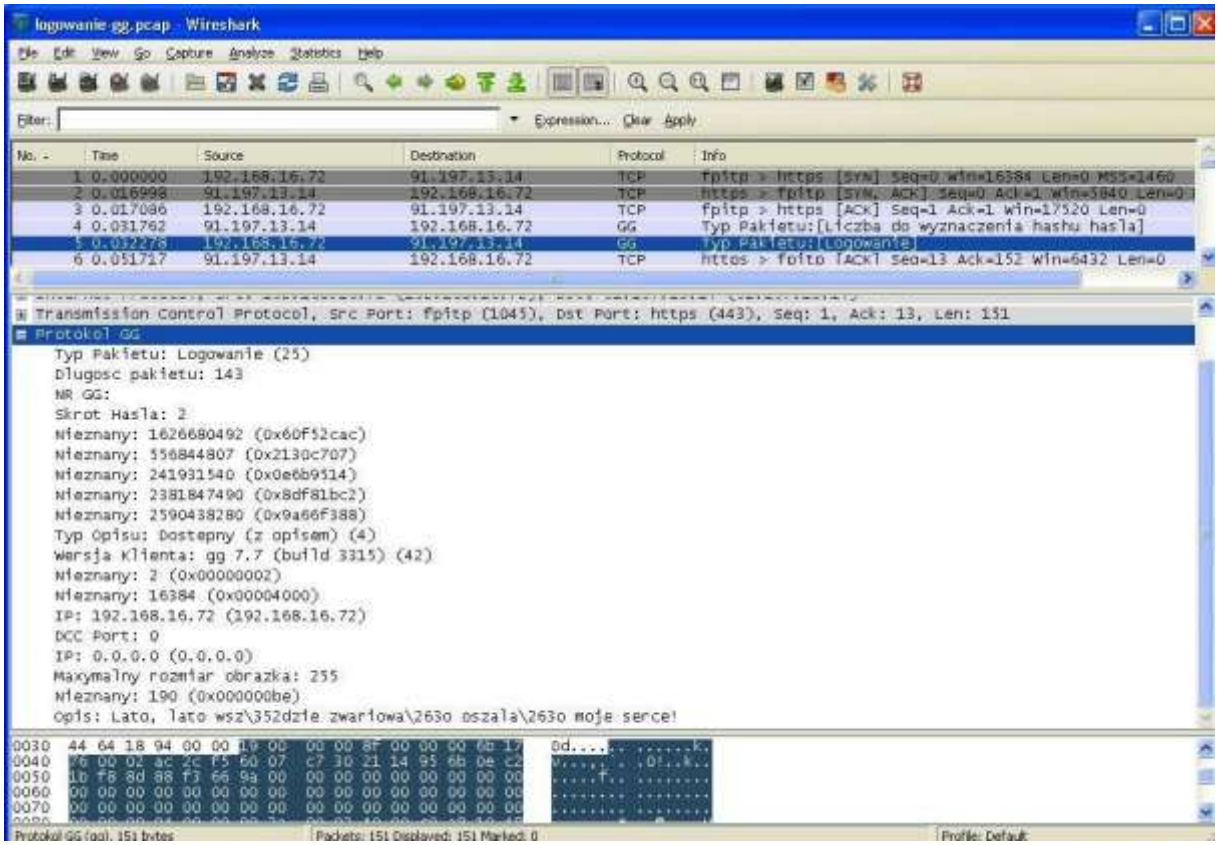
Opis instalacji oraz plik z wtyczką można znaleźć na stronie http://www.wireshark-gg.xt.pl/

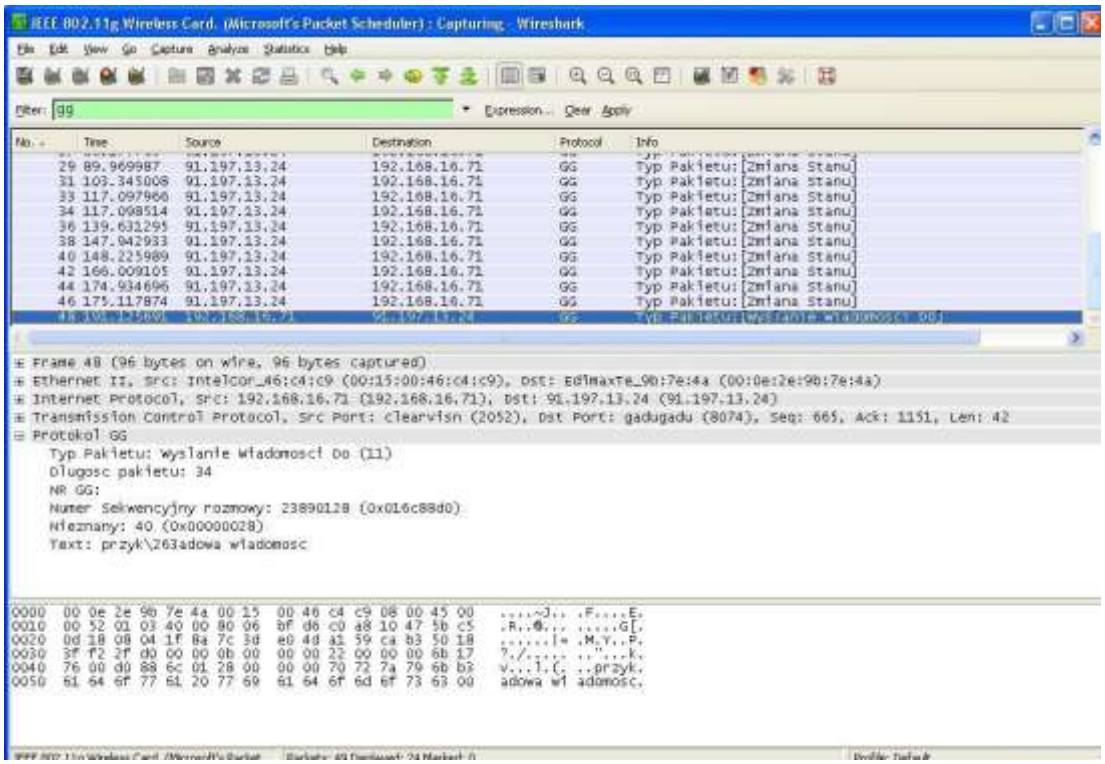
Po skopiowaniu pliku GG.dll do katalogu plugins/ restartujemy Wiresharka.

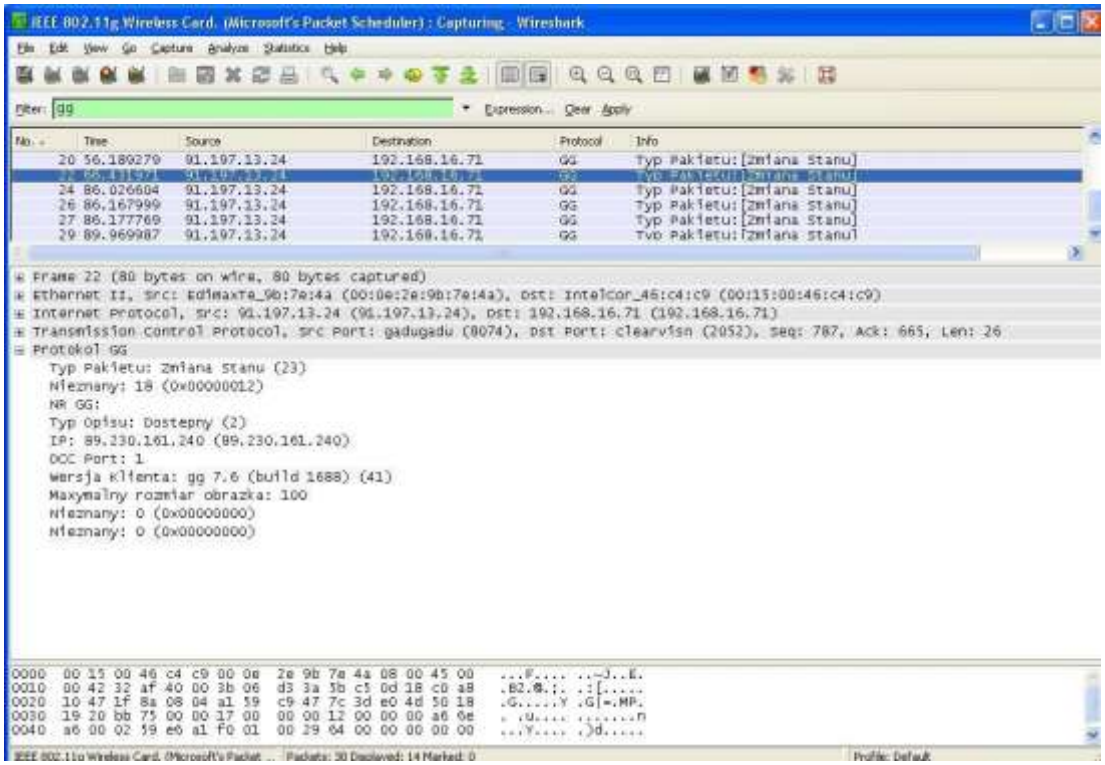
Do okna filtrów wpisujemy

```
gg
```

a następnie obserwujemy pakiety z żądaniami logowania, wysyłanymi wiadomościami, a także zmianami statusów.




IEEE 802.11g Wireless Card. (Microsoft's Packet Scheduler) : Capturing - Wireshark
File Edit View Go Capture Analyze Statistics Help
Filter: gg
Expression... Clear Apply
No. Time Source Destination Protocol Info
29 89.969987 91.197.13.24 192.168.16.71 GG Typ Pakietu:[Zmiana Stanu]
31 103.345008 91.197.13.24 192.168.16.71 GG Typ Pakietu:[Zmiana Stanu]
33 117.097966 91.197.13.24 192.168.16.71 GG Typ Pakietu:[Zmiana Stanu]
34 117.098514 91.197.13.24 192.168.16.71 GG Typ Pakietu:[Zmiana Stanu]
36 139.631295 91.197.13.24 192.168.16.71 GG Typ Pakietu:[Zmiana Stanu]
38 147.942933 91.197.13.24 192.168.16.71 GG Typ Pakietu:[Zmiana Stanu]
40 148.225989 91.197.13.24 192.168.16.71 GG Typ Pakietu:[Zmiana Stanu]
42 166.009105 91.197.13.24 192.168.16.71 GG Typ Pakietu:[Zmiana Stanu]
44 174.934696 91.197.13.24 192.168.16.71 GG Typ Pakietu:[Zmiana Stanu]
46 175.117874 91.197.13.24 192.168.16.71 GG Typ Pakietu:[Zmiana Stanu]
Frame 48 (96 bytes on wire, 96 bytes captured)
Ethernet II, Src: IntelCor_46:c4:c9 (00:15:00:46:c4:c9), Dst: EdimaxTe_9b:7e:4a (00:0e:2e:9b:7e:4a)
Internet Protocol, Src: 192.168.16.71 (192.168.16.71), Dst: 91.197.13.24 (91.197.13.24)
Transmission Control Protocol, Src Port: clearvisn (2052), Dst Port: gadugadu (8074), Seq: 665, Ack: 1151, Len: 42
Protokol GG
Typ Pakietu: Wysłanie Wiadomosci Do (11)
Dlugosc pakietu: 34
NR GG:
Numer Sekwencyjny rozmowy: 23890128 (0x016c88d0)
Nieznany: 40 (0x00000028)
Text: przyk\263adowa wiadomosc
0000 00 0e 2e 9b 7e 4a 00 15 00 46 c4 c9 08 00 45 00 ....~J... .F....E.
0010 00 52 01 03 40 00 80 06 bf d6 c0 a8 10 47 5b c5 .R..@... .....G[.
0020 0d 18 08 04 1f 8a 7c 3d e0 4d a1 59 ca b3 50 18 ......|= .M.Y..P.
0030 3f f2 2f d0 00 00 0b 00 00 00 22 00 00 00 6b 17 ?./..... .."...k.
0040 76 00 d0 88 6c 01 28 00 00 00 70 72 7a 79 6b b3 v...l.(. ..przyk.
0050 61 64 6f 77 61 20 77 69 61 64 6f 6d 6f 73 63 00 adowa wi adomosc.

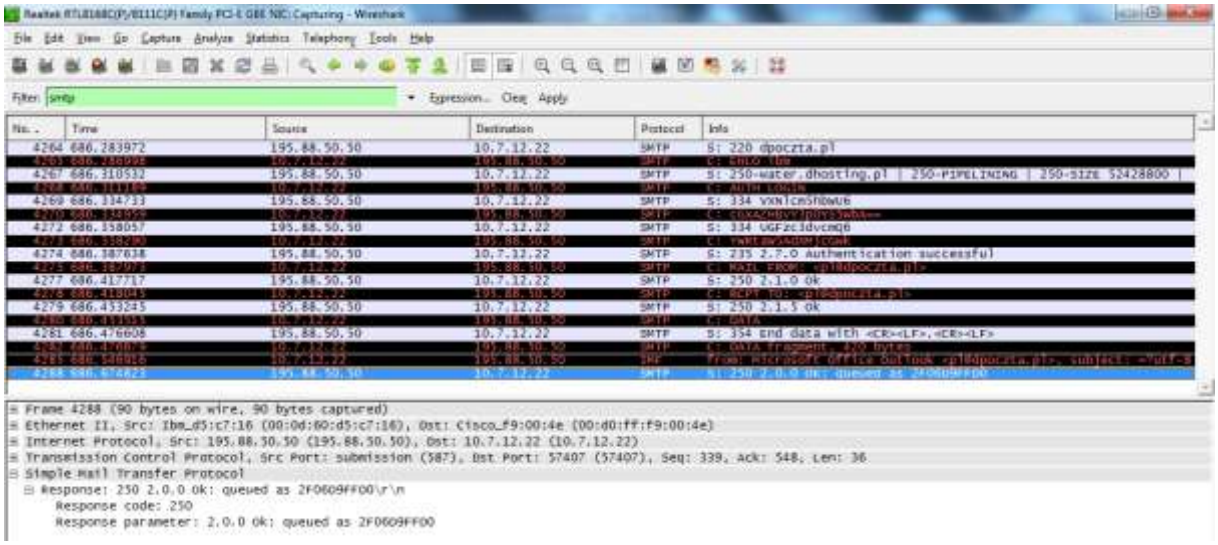

IEEE 802.11g Wireless Card. (Microsoft's Packet Scheduler) : Capturing - Wireshark
File Edit View Go Capture Analyze Statistics Help
Filter: gg
Expression... Clear Apply
No. Time Source Destination Protocol Info
20 56.189279 91.197.13.24 192.168.16.71 GG Typ Pakietu:[Zmiana Stanu]
24 86.026604 91.197.13.24 192.168.16.71 GG Typ Pakietu:[Zmiana Stanu]
26 86.167999 91.197.13.24 192.168.16.71 GG Typ Pakietu:[Zmiana Stanu]
27 86.177769 91.197.13.24 192.168.16.71 GG Typ Pakietu:[Zmiana Stanu]
29 89.969987 91.197.13.24 192.168.16.71 GG Typ Pakietu:[Zmiana Stanu]
Frame 22 (80 bytes on wire, 80 bytes captured)
Ethernet II, Src: EdimaxTe_9b:7e:4a (00:0e:2e:9b:7e:4a), Dst: IntelCor_46:c4:c9 (00:15:00:46:c4:c9)
Internet Protocol, Src: 91.197.13.24 (91.197.13.24), Dst: 192.168.16.71 (192.168.16.71)
Transmission Control Protocol, Src Port: gadugadu (8074), Dst Port: clearvisn (2052), Seq: 787, Ack: 665, Len: 26
Protokol GG
Typ Pakietu: Zmiana Stanu (23)
Nieznany: 18 (0x00000012)
NR GG:
Typ Opisu: Dostepny (2)
IP: 89.230.161.240 (89.230.161.240)
DCC Port: 1
Wersja Klienta: gg 7.6 (build 1688) (41)
Maxymalny rozmiar obrazka: 100
Nieznany: 0 (0x00000000)
Nieznany: 0 (0x00000000)
0000 00 15 00 46 c4 c9 00 0e 2e 9b 7e 4a 08 00 45 00 ...F.... ..~J..E.
0010 00 42 32 af 40 00 3b 06 d3 3a 5b c5 0d 18 c0 a8 .B2.@.;. .:[.....
0020 10 47 1f 8a 08 04 a1 59 c9 47 7c 3d e0 4d 50 18 .G.....Y .G|=.MP.
0030 19 20 bb 75 00 00 17 00 00 00 12 00 00 00 a6 6e . .u.... ....n
0040 a6 00 02 59 e6 a1 f0 01 00 29 64 00 00 00 00 00 ...Y.... .)d.....

4) Poczta elektroniczna – SMTP

Kolejnym naszym krokiem jest analiza listów elektronicznych wysyłanych przez nasz komputer. Możemy dzięki temu np. wyeliminować różnorakie programy wysyłające SPAM.

W tym celu ustawiamy filtr

```
smtp
```



Widzimy całą historię komunikacji z serwerem pocztowym. Jednak, gdy wysyłanych wiadomości jest kilka równolegle, analiza poszczególnych kawałków może być trudna.

Jeśli chcemy wyfiltrować wstępnie tylko źródłowe adresy e-mail, z których wysyłane są wiadomości stosujemy filtr

```
smtp.req.command == "MAIL" and
    smtp.req.parameter contains "FROM"
```

Wyświetlą nam się wszystkie pakiety zawierające fazę MAIL FROM sesji SMTP.

Jeśli zaś interesują nas odbiorcy wiadomości, możemy ustawić filtr

```
smtp.req.command == "RCPT"
```

Jego rezultat to wyświetlenie wszystkich pakietów RCPT TO sesji SMTP.

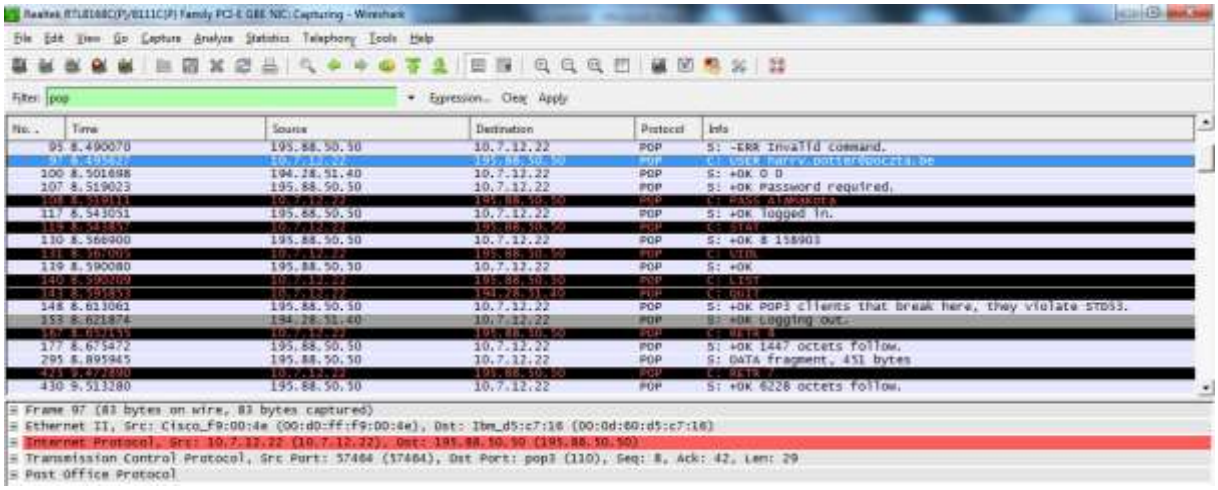
4) Poczta elektroniczna - POP3

Drugą częścią analizy poczty jest analiza ruchu poczty odbieranej z naszego serwera pocztowego – serwera Post Office Protocol.

a) wiadomości

Na początek ustawmy filtr główny dla POP3

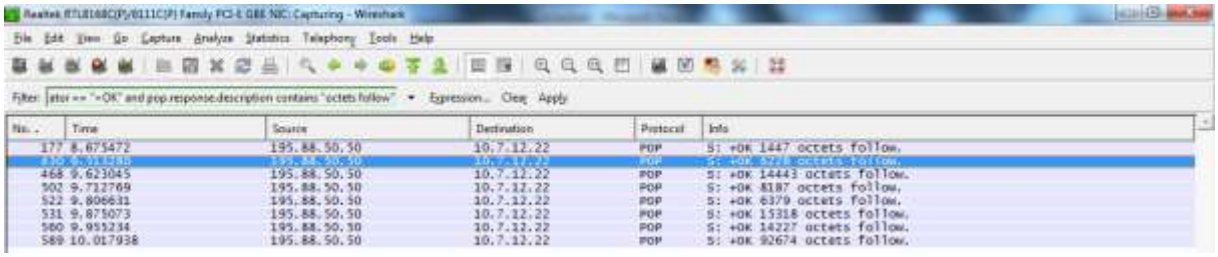
```
pop
```



Widzimy tu cały przebieg sesji z serwerem POP3.

Aby wyfiltrować jedynie wiadomości, wpisujemy jako filtr

```
pop.response.indicator == "+OK"
    and pop.response.description contains "octets"
```



Widzimy tu listę pakietów z pobieranymi przez klienta wiadomościami.

b) hasło

Jeśli zainteresowani jesteśmy przechwyceniem jawnego hasła, stosujemy

```
    pop.request.command == "PASS"
```

Na liście zauważamy wszystkie podane przez nas hasła dostępowe w zupełnie jawnej postaci.

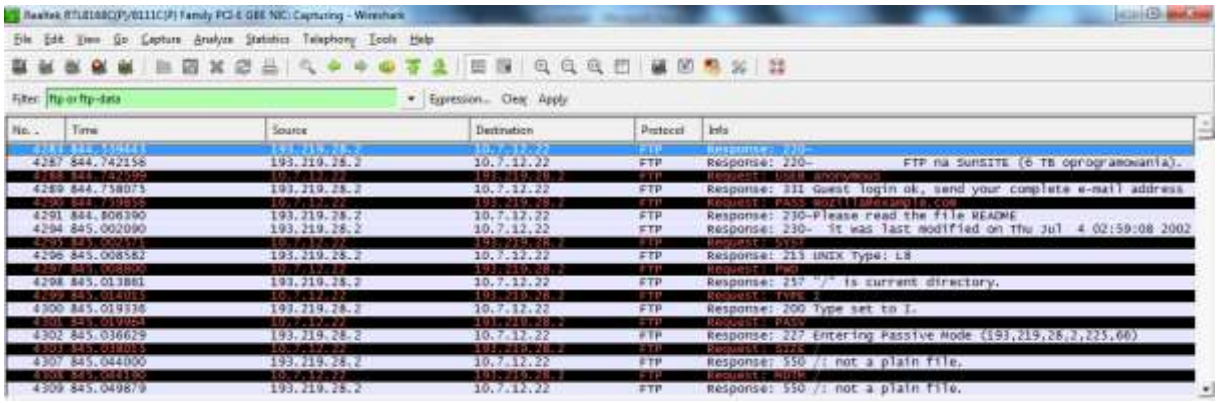
5) Przesył plików FTP

Ostatnim z analizowanych protokołów nie zakładających szyfrowania jest FTP. Służy on do zdalnego przesyłu plików. Zaobserwować możemy brak zabezpieczeń zarówno przesyłanych danych, jak i też nawet autoryzacji.

a) dane

Na wstępie przechwyćmy cały ruch FTP filtrem

```
ftp or ftp-data
```



Widzimy dużo informacji odnośnie całej sesji – wstępne uwierzytelnienie, wyświetlanie listy plików w katalogach, historię zmian katalogów przez użytkownika, oraz same dane.

Jeśli interesują nas nazwy pobieranych plików, stosujemy

```
ftp.request.command == "RETR"
```

Prezentowane będą wtedy tylko pakiety z nakazem pobrania zasobów.

Jeśli zaś szukamy wysyłanych z naszej sieci plików, pomocny będzie filtr

```
ftp.request.command == "STOR"
```

b) hasło

Protokół FTP także nie szyfruje w żaden sposób hasła. Aby poznać pary login/hasło, stosujemy

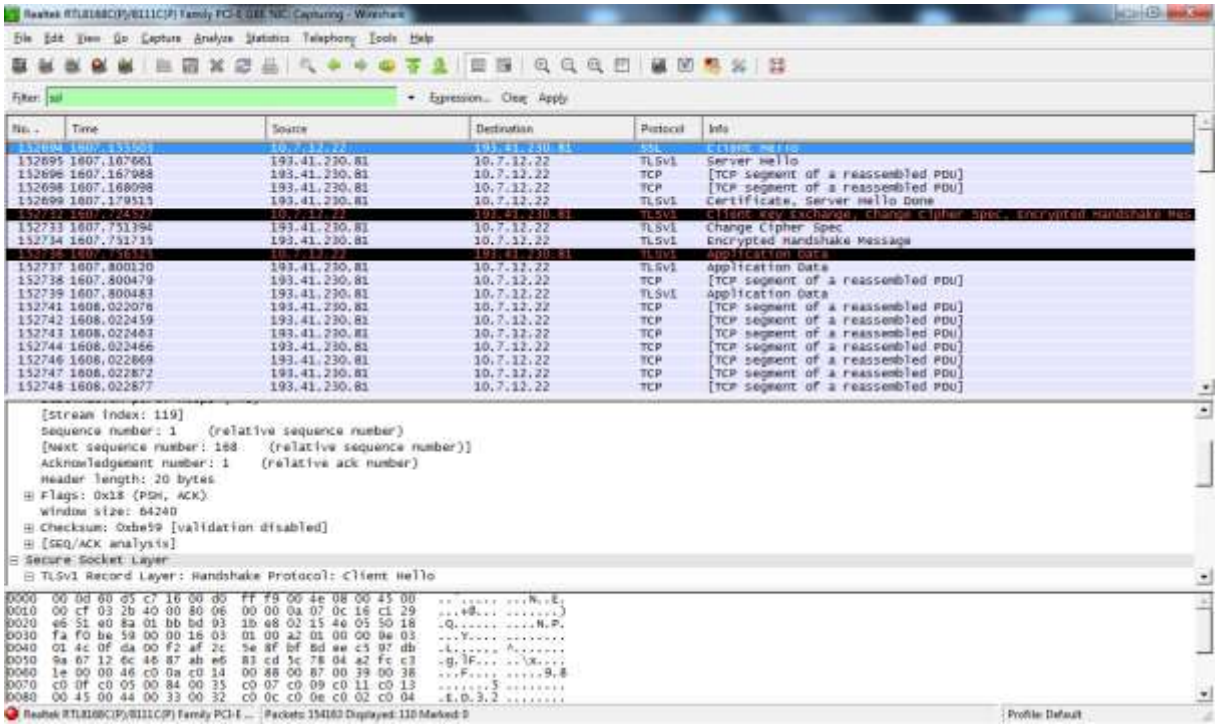
```
ftp.request.command == "USER" or ftp.request.command == "PASS"
```

Zwracane są pary wpisów – jedna z nazwą użytkownika, druga – z hasłem.

6) Transmisja HTTPS

Na samym końcu spróbujemy podsłuchać transmisję szyfrowaną przy użyciu protokołu SSL. W tym celu włączamy filtr wyświetlania

```
ssl
```



Widzimy, że żadnych użytecznych danych niestety jawnie nie widać.